УДК 004.042

Нижников Николай Юрьевич
Амурский государственный университет
г. Благовещенск, Россия
E-mail: sss_311771@mail.ru
Труфанова Татьяна Венаминовна
Амурский государственный университет
г. Благовещенск, Россия
E-mail: tvtr@mail.ru
Nizhnikov Nikolai Yurievich
Amur State University
Blagoveschensk, Russia
E-mail: sss_311771@mail.ru
Trufanova Tatiana Veniaminovna
Amur State University
Blagoveschensk, Russia
E-mail: tvtr@mail.ru

АВТОМАТИЗАЦИЯ ПРОЦЕССА ПЕРЕКЛЮЧЕНИЯ БАЗОВОЙ СТАНЦИИ СОТОВОЙ СВЯЗИ НА РЕЗЕРВНУЮ ВЫШКУ В СВЯЗИ С НЕИСПРАВНОСТЬЮ

AUTOMATION OF THE PROCESS OF SWITCHING A CELLULAR BASE STATION TO A BACKUP TOWER DUE TO A FAILURE

Аннотация. Работа посвящена автоматизации процесса переключения базовой станции сотовой связи на резервную вышку в связи с неисправностью с помощью кворума в сетевом трафике станции.

Abstract. This work is devoted to automating the process of switching a cellular base station to a backup tower due to a malfunction, using a quorum in the station's network traffic.

Ключевые слова: PostgreSQL (объектно-реляционная система управления базами данных – СУБД), кворум, резервные вышки.

Key words: PostgreSQL (object-relational database management system – DBMS), quorum, backup towers.

Введение

В статье предлагается разработка программного обеспечения для автоматизации процесса переключения базовой станции сотовой связи на резервную вышку для управления сетью и оборудованием в случае возникновения неполадок на базовой станции, что обеспечит удаленное взаимодействие оператора с инженерными системами и технологическими процессами.

Разработка программного обеспечения автоматического переключения на резервную

базовую станцию сотовой связи. Во время поиска сети смартфон вычисляет самый сильный и стабильный сигнал со всех окружающих его вышек и подключается к нему. Обычно это самая ближайшая точка. В на-стоящее время при неисправности или каких-либо проблемах на базовой станции нет возмож-ности автоматического переключения на резервную вышку с целью обеспечения непрерывности связи для абонентов.

Работа вышки сотовой связи обычно отслеживается с помощью системы мониторинга, которая помогает автоматически определить поломки или проблемы в работе. Когда система базовой станции обнаруживает такую проблему (например, выход из строя оборудования или потеря сигнала), она передает эту информацию в центр управления сетью (коммутатор).

В случае обнаружения поломки на базовой станции система автоматически активирует резервную вышку, которая располагается в том же районе или прилегающей к нему зоне обслуживания. Если такой вышки нет, то другие базовые станции данного оператора сотовой связи перенастраиваются для обеспечения дополнительного покрытия.

Переключение вышки сотовой связи в случае поломки называется процедурой перераспределения трафика, или переключения на резервную вышку (рис. 1) [1].

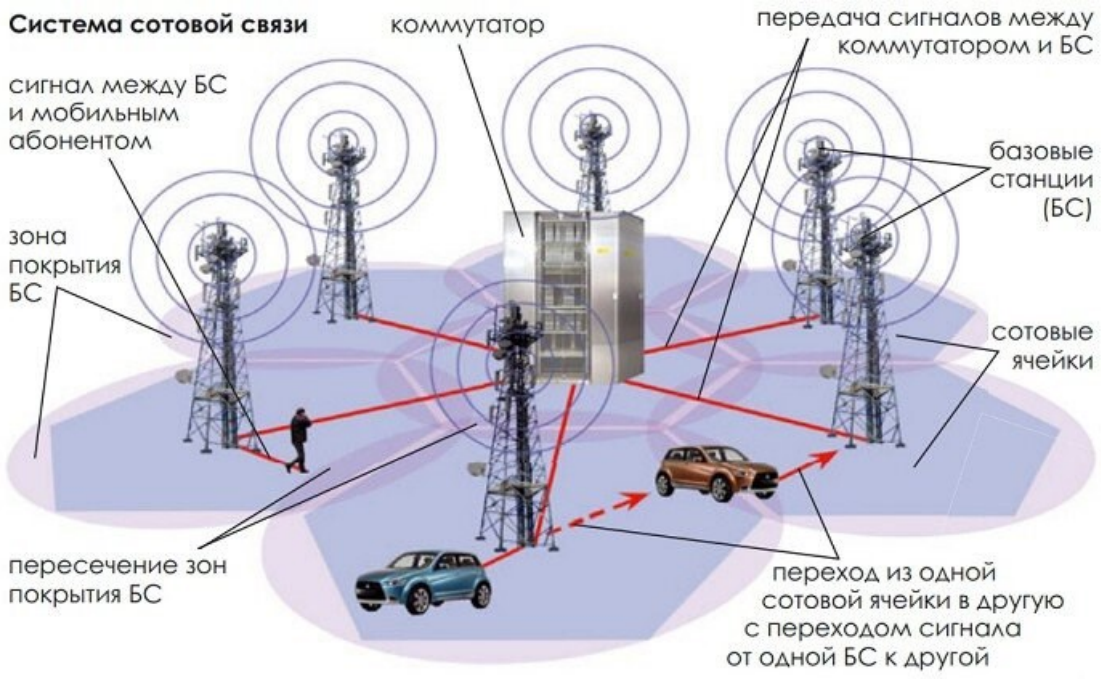

Рис. 1. Схема сотовой сети связи.

Когда резервная базовая станция активируется, сеть начинает перенаправлять трафик от вышки с поломкой на резервную [2]. Это может быть достигнуто путем изменения настроек сети или перенастройки направления сигнала. После активации резервной станции сетевые инженеры проводят тестирование с целью контроля обеспечения качества покрытия базовой станцией. Они продолжают мониторинг сети, чтобы убедиться, что всё работает корректно.

После того как поломка устранена, а основная вышка восстановлена, базовая станция

может вернуться к нормальному режиму работы. Возможно, потребуется некоторое время на переход обратно на основную вышку, чтобы убедиться в стабильности работы.

Автоматизируем процесс переключения на резервную вышку, чтобы минимизировать время простоя и обеспечить непрерывность обслуживания пользователей сети.

Кворум в сети станций трафика является критическим элементом для гарантирования надежности и стабильности системы. Если станции трафика не удастся связаться или увидеть другие станции, она может перестать работать и отказать в обслуживании клиентов.

В таком случае системы станции трафика включают в себя механизмы самодиагностики и восстановления. Если станция обнаруживает, что она не функционирует должным образом или не справляется с текущей нагрузкой, она может запросить помощь у соседних станций.

Соседние станции, в свою очередь, могут забрать часть трафика, что обеспечит непрерывное обслуживание клиентов, минимизирует задержки и предотвратит потерю данных. Таким образом, кворум играет важную роль в гарантировании эффективности и стабильности работы сети станций трафика. Благодаря кворуму сеть может быстро адаптироваться к изменениям и обеспечить надежное обслуживание клиентов, даже при возникновении проблем или сбоев.

В формализации кворума для системы станций трафика учитываются различные факторы, включая общее количество станций, текущую нагрузку, статус каждой станции и другие параметры.

Определяем кворум как минимальное количество работающих станций, необходимых для обеспечения надежной работы системы. Если общее количество станций в системе равно N, то кворум Q может быть определен как:

$Q = N / 2 + 1.$

Это означает, что для надежной работы системы требуется, чтобы работала хотя бы половина станций плюс одна.

С целью учета нагрузки на каждую станцию определим в кворуме как минимальное то суммарное количество баллов нагрузки, которое должно быть обработано работающими станциями. Если каждая станция имеет максимальную нагрузку L, то кворум Q может быть определен как:

$Q = N * L / 2 + 1.$

В этом случае кворум учитывает не только количество станций, но и их нагрузку.

Описание программы на Python [3], которая может быть использована для расчета кворума и сохранения результата, – в PostgreSQL [4]. Эта программа предполагает, что есть таблица stations в PostgreSQL, где хранится информация о станциях и о количестве работающих станций;

```
python
import psycopg2
from psycopg2 import sql
# Установить соединение с PostgreSQL
conn = psycopg2.connect(database="your_database", user="your_username", pass-
```

```
word="your_password", host="localhost", port="5432")
    # Создать курсор для выполнения SQL-запросов
    cur = conn.cursor()
    # Получить общее количество станций и количество работающих станций
    cur.execute("SELECT COUNT(), COUNT() FILTER (WHERE is_on = TRUE) FROM stations;")
    total_stations, working_stations = cur.fetchone()
    # Вычислить кворум
    quorum = total_stations / 2 + 1
    # Проверить, достигнут ли кворум
    if working_stations >= quorum:
       status = "Quorum reached"
    else:
       status = "Quorum not reached"
    # Сохранить результат в новой таблице в PostgreSQL
    cur.execute(
       sql.SQL("CREATE TABLE IF NOT EXISTS quorum_status (status text);"))
    cur.execute(
       sql.SQL("TRUNCATE TABLE quorum_status;"))
    cur.execute(
       sql.SQL("INSERT INTO quorum_status (status) VALUES (%s);"), [status])
    # Выполнить транзакцию и закрыть соединение
    conn.commit()
    cur.close()
    conn.close()
```

Программа сначала устанавливает соединение с PostgreSQL, получает общее количество станций и число работающих станций из таблицы station, затем вычисляет кворум. Результат сохраняется в новой таблице quorum_status в PostgreSQL.

Для этого нужно создать сценарий для каждой станции, который будет периодически отправлять информацию о своем состоянии в базу данных. Этот скрипт будет схожим с предыдущим, но с включением логики для определения состояния станции.

Предположим, что у вас есть функция check_station_status(), которая возвращает True, если станция включена, и False в противном случае. Тогда скрипт выглядит следующим образом:

```
    python
    import psycopg2
    from psycopg2 import sql
    def check_station_status():
       # Здесь должна быть вставка для проверки статуса станции
       pass
    # Установить соединение с PostgreSQL
```

```
conn = psycopg2.connect(database="your_database", user="your_username", password="your_password", host="localhost", port="5432")
# Создать курсор для выполнения SQL-запросов
cur = conn.cursor()
# Получить статус станции
is_on = check_station_status()
# Сохранить статус станции в таблице в PostgreSQL
cur.execute(
  sql.SQL("UPDATE stations SET is_on = %s WHERE station_id = %s;"), [is_on, "your_station_id"])
# Закоммитить транзакцию и закрыть соединение
conn.commit()
cur.close()
conn.close()
```

В этом скрипте предполагается, что в таблице stations есть столбец is_on, который указывает, включена ли станция, и столбец station_id, идентифицирующий каждую станцию. В работе нужно заменить "your_station_id" на идентификатор базовой станции.

Функция check_station_status() является заглушкой, которая заменяется на дополнения, необходимые для проверки статуса станции.

Заключение

В нашей работе реализована программа автоматического переключения базовых станций сотовой связи на резервные вышки в случае неисправности оборудования и перегрузки в сети. В предложенном методе используются кворум в сети станций трафика, что является критическим элементом и служит для проверки надежности и стабильности системы. Использование данной программы поможет автоматизировать работу вышек по выходу из критических ситуаций для обеспечения непрерывной связи абонентов.

1. Электронный источник. Сотовые, транковые и ячеистые системы мобильной связи [https://dzen.ru/a/ZB1hXGg1QRFh2aC8]

2. Кожемякина, М.М. Анализ пропускной способности и индикаторов качества базовой станции GSM мобильного оператора «Билайн» //Вестник магистратуры, 2018 [https://cyberleninka.ru/article/n/analiz-propusknoy-sposobnosti-i-indikatorov-kachestva-bazovoy-stantsii-gsm-mobilnogo-operatora-bilayn?ysclid=lu7yhx18qa255617936]

3. Еремин, И.Е., Еремина, В.В., Жилиндина, О.В. Базы данных. Учебно-методическое пособие. – Благовещенск: Амурский гос. ун-т, 2021 – 112 с.

4. PostgreSQL [Электронный ресурс][https://blog.skillfactory.ru/glossary/postgresql/]